



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/733,674	12/08/2000	Akira Tsuboi	1503.64973	2710

24978 7590 11/30/2005

GREER, BURNS & CRAIN
300 S WACKER DR
25TH FLOOR
CHICAGO, IL 60606

EXAMINER

STEELMAN, MARY J

ART UNIT

PAPER NUMBER

2191

DATE MAILED: 11/30/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/733,674

Applicant(s)

TSUBOI, AKIRA

Examiner

Mary J. Steelman

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 September 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 5,9,11,13,15 and 17-29 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☐ Claim(s) 5,9,11,13,15 and 17-29 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 08 December 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: _____

DETAILED ACTION

1. This Office Action is in response to Amendments and Remarks received 8 September 2005. Per Applicant's request, claim 6 has been canceled. Claims 5, 9, 11, 13, 15, and 17 have been amended. Claims 5, 9, 11, 13, 15, 17-29 are pending.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 15, 17, 22, 23, 28, and 29 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 15 and 17 do not clearly claim statutory embodiments. "A computer program embodied on a transmission medium..." (claim 15) and "A computer data signal embodied in a carrier wave containing a computer program used to direct a computer..." (claim 17) may be cured by inserting "tangibly" before "embodied."

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2191

4. Claims 5, 6, 9, 11, 13, 15, and 17-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,349,344 B1 to Sauntry et al., in view of US Patent 5,978,585 to Crelier, and further in view of US Patent 6,078,744 to Wolczko et al.

Per claims 5 and 9, Sauntry disclosed:

An apparatus having an execution unit for executing a machine language, compiling a source program into a machine language directly executable by the execution unit, and executing the machine language in a just-in-time-compiler system, comprising:

(Col. 3, lines 47-48, "The present invention describes devices, computers, computer-readable media, and systems of varying scope" (apparatus) and col. 10, lines 24-29, "whole the class files are being parsed during creation of the preload DLL file, JAVA byte code may also be compiled into native code, and this native code also stored in the DLL file. (compile source program into a machine language) This is comparable to the just-in-time (JIT) interpretation conducted on a typical JAVA virtual machine...")

-a storage unit storing for each function a machine language executable by the execution unit obtained by compiling a function described in the source program, and maintaining stored data after the source program has been executed;

(Col. 8, lines 33-35, "The file is desirably burned into ROM (or other nonvolatile storage device) (maintain stored data after execution) to create a run-time image of the JAVA class files ..." and

Art Unit: 2191

col. 7, lines 56-59, "The converter is desirably a software tool...that provides for the combination of class files into a single DLL file, where the DLL file is in portable executable (PE) format (store converted source into DLL in machine language)."

-compiling unit compiling the source program into a machine language executable by the execution unit;

(Col. 10, lines 24-27, "while the class files are being parsed during creation of the preload DLL file, JAVA byte code may also be compiled into native code (compile source code into machine language), and this native code also stored in the DLL file...time is not wasted later ...at run-time (execution).")

-an execution control unit instructing the execution unit to directly execute either a machine language compiled by said compiling unit or a machine language stored in said storage unit depending on a determination result obtained by said determination unit.

(Col. 8 , lines 52-61.)

Sauntry disclosed a storage control unit storing the machine language compiled by said compiling unit, but failed to disclose details regarding updating and the use of date and time.

However, Crelier disclosed:

-corresponding to update date and time of the source program compiled by said compiling unit";

(Abstract, lines 8-9, "Examination of the timestamps leads to detection of those files which have been modified. The system will recompile..." Also, col. 3, lines 26-34, "...system keeps track of several pieces of information...timestamps of sources and compiled files...Examination of the timestamps leads to detection of those files which have been modified...")

-a determination unit determining whether or not the update date and time of the source program matches an update date and time corresponding to the machine language stored in said storage

(Abstract, lines 10-13, "The system will recompile A.java in the following circumstances: (1) A.class is not found, (2) A.java has a different timestamp, or (3) A.class has a different timestamp." Also see FIGS. 4A-C and col. 10, lines 10-38, "...at step 402, the source file's timestamp is examined to determine whether it has changed...")

Sauntry / Crelie failed to specifically provide details related to the following newly added limitations shown below.

More specifically, Wolczko disclosed:

-a read unit reading a program file storing the source program, wherein:

-said storage control unit stores the machine language in said storage unit by assuming that the update date and time of the program file indicated in the program file is the update date and time of the source program corresponding to the machine language;

Wolczko (col. 2, lines 43-51) disclosed improving performance of a compiler by journaling compilation data. The journaled information is then used during subsequent compilations instead of recomputing... Col. 7, lines 18-22, "The record also distinguishes when the compilation unit has changed (update and time of the source program) between the initial and subsequent compilations.

-said determination unit determines whether or not the update date and time of the program file indicated in the program file matches the update date and time stored in said storage unit corresponding the machine language;

Wolczko: See FIG. 6, #611. A determination is made regarding equivalence of the program. Related text at col. 10, line 32-col. 11, line 22. Col. 11, line 20, "Those skilled in the art will also understand that these decisions can be heuristically programmed based on measured parameters. These measurements are made by instrumenting a compiler to gather information about the decisions made during each compilation phase and then compiling a sufficient number of representative programs. These measurements are then used to program the 'journal ICD' decision procedure 513 and the 'equivalent unit in journal' decision procedure 611 for a specific computer programming language and computing system environment."

-the stored data are maintained even after the source program has been executed, the execution control unit is able to execute the source program without a time lag caused in the program

Art Unit: 2191

compiling process when the execution is started by instructing the execution unit to directly execute the machine language of the function stored in the storage unit without waiting for the compilation of the source program by the compiling unit when the same source program is re-executed;

Wolczko: Col. 6, lines 57-61, "The instant invention improves the performance of compilers by journaling (stored data maintained even after source program has been executed) intermediate compiler data (ICD) generated during an initial compilation of the source program, for later reuse (reuse stored code rather than waiting for compilation if same source program is re-executed) by a subsequent compilation of the source program. (Reuse eliminates a 'time lag' caused in the program compiling process by instructing the execution unit to directly execute the stored data.) Additionally, (col. 7, lines 14-22) "the invention saves to a journal the ICDs generated during an initial compilation of the source program. The journaled information can be kept on the computer's filesystem (maintained even after the source program has been executed). Each journal record identifies the compilation unit related to the record. The record also distinguishes when the compilation unit has changed between the initial and subsequent compilation.

-when the source program executed later does not match that executed previously by an amendment after the update of the source program, the storage control unit instructs the storage unit to store the machine language compiled by the compiling unit corresponding to the update date and time of the source program compiled by the compiling unit,

Wolczko: See FIG. 4 & related text at col. 8, line 46-col. 9, line 60. An initial compilation process 401 and a subsequent compilation process 403 are shown. At 409 an initial compilation procedure 413 saves the ICD for the compiled compilation units. A subsequent compilation process 403 initiates and (col. 9, line 19-21) generates its own journal for use by compilations subsequent. See FIG. 6, #611. If the equivalent unit (does not match) is not in the Journal, at #613, new data is generated for the journal (store machine language compiled by the compiling unit corresponding to the update data and time).

-the determination unit determines whether or not the update date and time of the source program matches the update date and time stored in the storage unit corresponding to the machine language,

Wolczko: A determination is made regarding equivalence of the program. See FIG. 6, #611 and related text at col. 10, line 32-col. 11, line 22. Col. 11, line 20, "Those skilled in the art will also understand that these decisions can be heuristically programmed based on measured parameters (date and time). These measurements are made by instrumenting a compiler to gather information about the decisions made during each compilation phase and then compiling a sufficient number of representative programs. These measurements are then used to program the 'journal ICD' decision procedure 513 and the 'equivalent unit in journal' decision procedure 611 for a specific computer programming language and computing system environment."

Art Unit: 2191

-when they do not match each other as a result of the determination, the execution control unit instructs the execution unit to execute the machine language newly compiled by the compiling unit although the machine language obtained by compiling the function used in the source program is stored in the storage unit.

See FIG. 6, #611 and related text at col. 10, line 32-col. 11, line 22. Col. 11, line 20, "Those skilled in the art will also understand that these decisions (match decisions) can be heuristically programmed based on measured parameters (date and time). These measurements are made by instrumenting a compiler to gather information about the decisions made during each compilation phase and then compiling a sufficient number of representative programs. These measurements are then used to program the 'journal ICD' decision procedure 513 and the 'equivalent unit in journal' decision procedure 611 for a specific computer programming language and computing system environment." See FIG. 4 & related text at col. 8, line 46-col. 9, line 60. An initial compilation process 401 and a subsequent compilation process 403 are shown. At 409 an initial compilation procedure 413 saves the ICD for the compiled compilation units. A subsequent compilation process 403 initiates and (col. 9, line 19-21) generates its own journal for use by compilations subsequent. See FIG. 6, #611. If the equivalent unit (does not match) is not in the Journal, at #613, new data is generated for the journal (store machine language compiled by the compiling unit corresponding to the update data and time).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention, to have modified Sauntry's invention, to include information regarding timestamps

Art Unit: 2191

and program modification requiring recompilation, as provided by Crelrier, as this is a useful technique for maintaining logic regarding updates and to track versioning as software evolves. Timestamps are well known in the art for detecting versions and updates. Sauntry's invention relates (col. 1, lines 16-20) to "facilitating development of software programs, with particular emphasis on decreasing the time such a system spends on recompiling source modules..." and thus would be an obvious combination of arts.

It would have been further obvious, to one of ordinary skill in the art, at the time of the invention, to modify Sauntry / Crelrier, by using Wolczko to provide the missing features of using 'journaled compilation data' (col. 2, lines 65-67) in lieu of recreating datum during a subsequent compilation, if the equivalent unit is available in the journal, (FIG. 6, #611), because Wolczko also recognized the importance of (col. 1, line 12) 'increasing a compiler's performance' where (col. 2, lines 30-31) a compiler's improved performance may shorten the development cycle. By compiling only the portions of code that have been modified, and otherwise (col. 2, line 65-67) "using a first journaled datum in lieu of recreating an intermediate compilation datum during the subsequent compilation" compiler performance may be improved. One skilled in the art would understand that many techniques (such as timestamps) exist for determining equivalence of the compilation unit. Thus, it should be noted that all references depict the state of the art in 1997, whereby storing precompiled code and comparing equivalence to a possibly modified later code segment, was known in the art.

Per claims 11, 13, 15, and 17, Sauntry disclosed:

Art Unit: 2191

A method for executing a program based on a just-in-time-compiler system for compiling a source program into a machine language directly executable on a platform of a specific processing system, and executing the machine language, comprising:

(Col. 8, lines 47-52, "At run-time (executing), the JAVA virtual machine does a LoadLibrary call and a GetProcAddress call..." and col. 9, lines 23-26, "This method is inclusive of the steps or acts required to be taken by a device such as a computer to preload and preparse at least one JAVA class file into a run-time image (machine language) stored on a nonvolatile storage device such as a ROM." Also col. 10, lines 26-27, "This is comparable to the just-in-time (JIT) interpretation ...")

-storing the machine language obtained by compiling the source program for each function described in the source program;

(Col. 10, lines 23-26, "...while the class files are being parsed during creation of the preload DLL file, JAVA byte code may also be compiled into native code, and this native code also stored in the DLL file.")

Sauntry disclosed a storage control unit stores the machine language in said storage unit, but failed to disclose details regarding updating and the use of date and time.

However, Crelier disclosed:

Art Unit: 2191

-storing compiled code corresponding to an update date and time of the source program before compiled into a machine language;

(Col. 3, lines 29-31, “the system keeps track of several pieces of information. From the outset, the system has kept track of the timestamps of sources...and compiled files thereof...”)

-determining whether or not the date and time of the update of the source program matches an update date and time corresponding to the stored machine language;

(Col. 3, lines 33-34, “Examination of the timestamps leads to detection of those files which have been modified.”)

-setting either the machine language obtained by compiling the source program or the machine language stored in the storage unit to be directly executed on a platform of a specific processing system based on a determination result;

(Col. 3, lines 34-47, “The system will recompile...in the following circumstances...Otherwise, the system does not invoke a recompile...”)

Sauntry / Crelier failed to specifically disclose the following newly added limitations.

However, Wolczko disclosed:

Art Unit: 2191

- reading, with a read unit, a program file storing the source program, wherein:
- a storage control unit stores the machine language in a storage unit by assuming that the update date and time of the program file indicated in the program file is the update date and time of the source program corresponding to the machine language;
- a determination unit determines whether or not the update date and time of the program file indicated in the program file matches the update date and time stored in the storage unit corresponding the machine language;
- the stored data are maintained event after the source program has been execute, and execution control unit is able to execute the source program without a time lag caused in the program compiling process when the execution is started by instructing the execution unit to directly execute the machine language of the function stored in the storage unit without waiting for the compilation of the source program by a compiling unit when the same source program is re-executed;
- when the source program executed later does not match that executed previously by an amendment after the update of the source program, the storage control unit instructs the storage unit to store the machine language compiled by the compiling unit corresponding to the update date and time of the source program compiled by the compiling unit, the determination unit determines whether or not the update date and time of the source program matches the update date and time stored in the storage unit corresponding to the machine language,
- when they do not match each other as a result of the determination, the execution control unit instructs the execution unit to execute the machine language newly compiled by the compiling

Art Unit: 2191

unit although the machine language obtained by compiling the function used in the source program is stored in the storage unit.

(See rejection of limitations as addressed in claim 5 above.)

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention, to have modified Sauntry's invention, to include information regarding timestamps and program modification requiring recompilation, as provided by Crelier, as this is a useful technique for maintaining logic regarding updates and to track versioning as software evolves. Timestamps are well known in the art for detecting versions and updates. Sauntry's invention relates (col. 1, lines 16-20) to "facilitating development of software programs, with particular emphasis on decreasing the time such a system spends on recompiling source modules..." and thus would be an obvious combination of arts.

It would have been further obvious, to one of ordinary skill in the art, at the time of the invention, to modify Sauntry / Crelier, by using Wolczko to provide the missing features of using 'journaled compilation data' (col. 2, lines 65-67) in lieu of recreating datum during a subsequent compilation, if the equivalent unit is available in the journal, (FIG. 6, #611), because Wolczko also recognized the importance of (col. 1, line 12) 'increasing a compiler's performance' where (col. 2, lines 30-31) a compiler's improved performance may shorten the development cycle. By compiling only the portions of code that have been modified, and otherwise (col. 2, line 65-67) "using a first journaled datum in lieu of recreating an intermediate compilation datum during the subsequent compilation" compiler performance may be improved. One skilled in the art would

Art Unit: 2191

understand that many techniques (such as timestamps) exist for determining equivalence of the compilation unit. Thus, it should be noted that all references depict the state of the art in 1997, whereby storing precompiled code and comparing equivalence to a possibly modified later code segment, was known in the art.

Per claims 18-23:

-the stored machine language is stored in RAM.

Sauntry disclosed (col. 7, lines 19-20), “the converter loads and parses the Java class files into a file, which is then desirably burned into ROM...”(col. 8, lines 33-35), “The file is desirably burned into ROM (or other nonvolatile storage device) (maintain stored data after execution) to create a run-time image of the JAVA class files ...” It is inherent that the file is stored in RAM prior to burning into ROM. It is well known that a file may be brought into RAM at the time the stored code is to be executed to increase execution speed.

Per claims 24-29:

-storage unit also has stored thereon a standard source program compiled from an original source program.

Sauntry disclosed, (col. 3, lines 5-6), “the run-time image is a DLL file (standard source program, compiled from original source program) stored in read-only memory (ROM)...”, col. 7, lines 55-65, “The converter is desirably a software tool...that provides for the combination of

Art Unit: 2191

class files into a single DLL file, where the DLL file is in portable executable (PE) (standard source program, compiled from original source program) format known in the art...”

Response to Arguments

5. Applicant's arguments with respect to claims 5, 9, 11, 13, 15, and 17-29 have been considered but are moot in view of the new grounds of rejection.

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

11/21/2005



WEI Y. ZHEN
PRIMARY EXAMINER